CMPUT 291 Fall Term 2019
Mini Project 2 Report

Partner 1: Mohit Kumar (kumar2)

Partner 2: Toluwani Adekunle (taadekun)

Partner 3: Luke Patrick Dela Cruz (lukepatr)

I - General Overview and User Guide

Run at the project root to enter our interface:

```
$ python3 main.py
```

Please enter the relative path to your XML data file next by following the prompts on-screen.

Then query indefinitely, type ‘exit’ to quit

```
Query:
```

II - Detailed Design

Design Decisions

- To simplify user experience all that’s required for them is to provide a relative path to their XML file and we will handle build indexes, and file transforms in the background
- The following search strategy is as follows:
    - Generate a set of IDs (from all sub-commands intersected) from te.idx (terms), em.idx (emails), and da.idx (dates) which are all b-trees.
    - Hit the re.idx, which is hashed, with this set of IDs to get all the XML records

III - Testing Strategy

General testing strategies:

- Check for performance (generate data set with >1k records)
    - Assumption: inverted list of a term can fit in main memory
- Check for consistency (compound as many combinations of sub commands as possible)
- Let other team members test code they didn’t write themselves to avoid confirmation bias

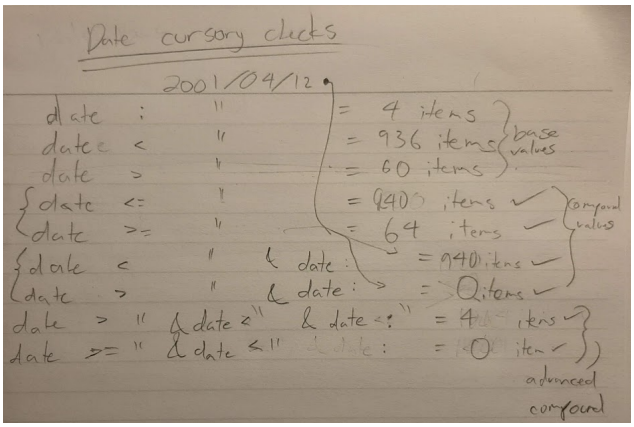
Most commonly encountered bugs:

- Intersections were failing/inconsistent because when two or more sub commands are present extra care must be used to ensure *that when at least two subIDs sets are intersected and return an empty the entire intersection set must also be empty (AND policy)*

- Date ranges not giving consistent results because <=, and >= were incorrectly thought of as intersections between < and =, but in fact is the **UNION** (reads are less/greater than **OR** equal to)

Portion specific testing cases/strategies:

- All files specific to testing the phases must either be in their respective folders or in the main directory i.e. 10.xml which is used for phase1 must either be in the “phase1src” folder or in the main directory
- Phase 1's speed for different number of entries (Upto 10 million) and correctness of output with 10.xml and 1k.xml given tested by phase1_test.py.
- Email lexer:
    - Must match: test@test.com.ca.us.gb.fr

- Dates test cases written out (using spec given data and the date “2001/04/12”:
    - 

IV - Group Work Break-down

Luke’s responsibilities:

- Command parser/lexer [~8 hours]
- Search architecture and design [~7.5 hours]
- Source code design and team coordination [~1.5 hours]
- Building index files from text files using db_load [~1 hours]

Toluwani’s responsibilities:

- Phase 2 building indexes and integrating the index builder in python [~1 hours]
- Tying phase 1 and phase 2 together in the main function and ensuring correctness of index files [~6 hours]
- Phase 3 testing [~1 hour]

- Error checking for both phase 1 and phase 2 [~2 hours]
- Organizing file directories in python [~1 hour]
- Formatting and manipulating output for phase 3 [~2 hours]

Mohit's responsibilities:

- Pase 1 xml parsing [~8 hours]
- Phase 1 test_class and bug fixing for correctness [~3 hours]
- Formatting and manipulating output for phase 3[~2hours]
- Phase 3 testing [~1hour]
- Phase 1 testing with bigger datafiles[1hour]
- Team coordination[1 hour]

We're all responsible for testing the entire project as a whole, and working on this design document. Our main method of coordination is using git version control alongside Github to keep track of commits, and feature pull request into the master project branch.